M 5

```
MMM         MMM     AAAAAAAAA        CCCCCCCCCCCC  RRRRRRRRRRRR        000000000
MMM         MMM     AAAAAAAAA        CCCCCCCCCCCC  RRRRRRRRRRRR        000000000
MMM         MMM     AAAAAAAAA        CCCCCCCCCCCC  RRRRRRRRRRRR        000000000
MMMMMM   MMMMMM  AAA         AAA  CCC              RRR         RRR  000         000
MMMMMM   MMMMMM  AAA         AAA  CCC              RRR         RRR  000         000
MMMMMM   MMMMMM  AAA         AAA  CCC              RRR         RRR  000         000
MMM   MMM   MMM  AAA         AAA  CCC              RRR         RRR  000         000
MMM   MMM   MMM  AAA         AAA  CCC              RRR         RRR  000         000
MMM   MMM   MMM  AAA         AAA  CCC              RRR         RRR  000         000
MMM         MMM  AAA         AAA  CCC              RRRRRRRRRRRR     000         000
MMM         MMM  AAA         AAA  CCC              RRRRRRRRRRRR     000         000
MMM         MMM  AAA         AAA  CCC              RRRRRRRRRRRR     000         000
MMM         MMM  AAAAAAAAAAAAAAA  CCC              RRR   RRR        000         000
MMM         MMM  AAAAAAAAAAAAAAA  CCC              RRR   RRR        000         000
MMM         MMM  AAAAAAAAAAAAAAA  CCC              RRR   RRR        000         000
MMM         MMM  AAA         AAA  CCC              RRR      RRR     000         000
MMM         MMM  AAA         AAA  CCC              RRR      RRR     000         000
MMM         MMM  AAA         AAA  CCC              RRR      RRR     000         000
MMM         MMM  AAA         AAA     CCCCCCCCCCCC  RRR         RRR     000000000
MMM         MMM  AAA         AAA     CCCCCCCCCCCC  RRR         RRR     000000000
MMM         MMM  AAA         AAA     CCCCCCCCCCCC  RRR         RRR     000000000
```

```
**FILE**ID**INPUT                                            M 15




    IIIIII    NN        NN  PPPPPPPP    UU        UU   TTTTTTTTTT
    IIIIII    NN        NN  PPPPPPPP    UU        UU   TTTTTTTTTT
      II      NN        NN  PP      PP  UU        UU       TT
      II      NN        NN  PP      PP  UU        UU       TT
      II      NNNN      NN  PP      PP  UU        UU       TT
      II      NNNN      NN  PP      PP  UU        UU       TT
      II      NN  NN    NN  PPPPPPPP    UU        UU       TT
      II      NN  NN    NN  PPPPPPPP    UU        UU       TT
      II      NN      NNNN  PP          UU        UU       TT
      II      NN      NNNN  PP          UU        UU       TT
      II      NN        NN  PP          UU        UU       TT
      II      NN        NN  PP          UU        UU       TT            ....
    IIIIII    NN        NN  PP          UUUUUUUUUU         TT            ....
    IIIIII    NN        NN  PP          UUUUUUUUUU         TT            ....

  LL              IIIIII      SSSSSSSS
  LL              IIIIII      SSSSSSSS
  LL                II      SS
  LL                II      SS
  LL                II      SS
  LL                II      SS
  LL                II        SSSSSS
  LL                II        SSSSSS
  LL                II              SS
  LL                II              SS
  LL                II              SS
  LL                II              SS
  LLLLLLLLLL      IIIIII    SSSSSSSS
  LLLLLLLLLL      IIIIII    SSSSSSSS
```

MA
VO4

Table of contents

```
0000            .TITLE  MAC$INPUT GET NEXT CHARACTER
0000            .IDENT  'V04-000'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000
0000    ;++
0000    ; FACILITY:     VAX MACRO ASSEMBLER OBJECT LIBRARY
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ; The VAX-11 MACRO assembler translates MACRO-32 source code into object
0000    ; modules for input to the VAX-11 LINKER.
0000    ;
0000    ; ENVIRONMENT:  USER MODE
0000    ;
0000    ; AUTHOR: Benn Schreiber, CREATION DATE: 21-AUG-78
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03.01  MTR0011         Mike Rhodes     18-Mar-1982
0000    ;               Add logic to routine MAC$GETLIN to trap non-RMS$_EOF errors
0000    ;               and non-SUM$_xxx errors.  This occurs when an illegal record
0000    ;               is encountered by RMS and it does not count as a SUM error,
0000    ;               which leaves us in an infinite loop trying to read the next line.
0000    ;               Fixes QAR #691.
0000    ;
0000    ;       V03.00  MTR0006         Mike Rhodes     15-Mar-1982
0000    ;               Modify MAC$GETCHR to allow ALL characters to be passed
0000    ;               when the FLG$V_DLIMSTR flag is set.  The characters allowed
0000    ;               to be passed include the semicolon (which was already passed)
0000    ;               and the hyphen, which at times was incorrectly treated as a
0000    ;               line continuation character.  Fixes SPR #11-42904.
0000    ;
0000    ;       V01.10  RN0023          R. Newland      3-Nov-1979
```

```
0000  ;                      New message codes to get error messages from system
0000  ;                      message file.
0000  ;
0000  ;              V01.09  RN0010          R. Newland      5-Sep-1979
0000  ;                      Mulitpage MXB blocks
0000  ;
0000  ;              V01.08  RN0005          R. Newland      27-Aug-1979
0000  ;                      Remove .ALIGN LONG statements
0000  ;
0000  ;              V01.07  RN0002          R. Newland      01-Feb-1979
0000  ;                      Changes for Source Update Merge, Get input lines
0000  ;                      from SUM$LINE.
0000  ;
0000  ;--
```

```
0000     73            .SBTTL  DECLARATIONS
0000     74    ;
0000     75    ; INCLUDE FILES:
0000     76    ;
0000     77
0000     78    ;
0000     79    ; MACROS:
0000     80    ;
0000     81            $RABDEF                         ;DEFINE RAB OFFSETS
0000     82            $MAC_GENVALDEF                  ;VAX-11 MACRO GENERAL SYMBOLS
0000     83            $MAC_CTLFLGDEF                  ;CONTROL FLAGS
0000     84            $MAC_INTCODDEF                  ;INTERMEDIATE CODE FILE SYMBOLS
0000     85            $MAC_INPBLKDEF                  ;DEFINE INPUT BLOCK OFFSETS
003C     86            $MAC_SYMBLKDEF                  ;DEFINE SYMBOL BLOCK OFFSETS
0000     87            $MAC_MNBDEF                     ;DEFINE MNB OFFSETS
0000     88            $MACMSGDEF                      ; Define message codes
0000     89            $STSDEF                         ;STATUS BITS
0000     90            DEFSUMCBL                       ; Define SUM control block symbols
0000     91    ;
0000     92
0000     93    ;
0000     94    ; EQUATED SYMBOLS:
0000     95    ;
0000     96
0000     97    ;
0000     98    ; OWN STORAGE:
0000     99    ;
0000    100
```

```
                          0000          .SBTTL  MAC$GETCHR GET NEXT CHARACTER FROM INPUT STREAM
                          0000    ;++
                          0000    ; FUNCTIONAL DESCRIPTION:
                          0000    ;
                          0000    ;         THIS ROUTINE IS CALLED WHENEVER A NEW CHARACTER IS NEEDED.
                          0000    ;         IT PERFORMS ALL THE HOUSEKEEPING FOR ENDS OF LINES, OUTPUTS
                          0000    ;         COMMANDS TO THE INT. FILE FOR NEW LINES, AND HANDLES CONTINUATION
                          0000    ;         LINES.
                          0000    ;
                          0000    ; CALLING SEQUENCE:
                          0000    ;
                          0000    ;         JSB     MAC$GETCHR
                          0000    ;
                          0000    ; INPUT PARAMETERS:
                          0000    ;
                          0000    ;         R10     LAST CHARACTER READ
                          0000    ;         R11     POINTS TO FLAGS (MAC$GL_FLAGS)
                          0000    ;                 IF FLG$M_ALLCHR IS SET THEN SEMICOLONS WILL BE PASSED
                          0000    ;                         BACK. IF THE FLAG IS CLEAR, SEMICOLONS AND
                          0000    ;                         EVERYTHING FOLLOWING WILL BE IGNORED.
                          0000    ;
                          0000    ; IMPLICIT INPUTS:
                          0000    ;
                          0000    ;         NONE
                          0000    ;
                          0000    ; OUTPUT PARAMETERS:
                          0000    ;
                          0000    ;         R10     NEXT CHARACTER
                          0000    ;
                          0000    ; IMPLICIT OUTPUTS:
                          0000    ;
                          0000    ;         NONE
                          0000    ;
                          0000    ; COMPLETION CODES:
                          0000    ;
                          0000    ;         NONE
                          0000    ;
                          0000    ; SIDE EFFECTS:
                          0000    ;
                          0000    ;         NONE
                          0000    ;
                          0000    ;--
                          0000
                          0000
                          0000
                      00000000          .PSECT  MAC$RO_CODE_P15,NOWRT,GBL,LONG
                          0000
                          0000          .ENABL  LSB
                          0000
                          0000    MAC$GETCHR::
             58     DD    0000          PUSHL   R8                      ;SAVE R8
   58   0000'CF     9E    0002    10$:  MOVAB   W^MAC$GL_LINEPT,R8      ;POINT TO MAC$GL_LINEPT
        0D   5A     91    0007    20$:  CMPB    R10,#CR                 ;TIME TO READ NEW LINE?
             1F     12    000A          BNEQ    30$                     ;IF NEQ NO
        5A   20     9A    000C    25$:  MOVZBL  #BLNK,R10               ;YES--PREVENT LOOPING ON CR
   50   0000'CF     D0    000F          MOVL    W^MAC$GL_INPUTP,R0      ;ADDRESS INPUT BLOCK
        6B   08     C8    0014          BISL2   #FLG$M_CONT,(R11)       ;ALLOW CONTINUATION LINES
```

```
                08 B0     16 0017          JSB     @INP$L_GETL(R0)              ;CALL INPUT ROUTINE TO READ NEXT LINE
           E9 6B    10    E0 001A          BBS     #FLG$V_MACTXT,(R11),20$      ;BRANCH IF READING MACRO TEXT
                             001E          $INTOUT_WD INT$_NEWL,-               ;YES--SIGNAL NEW LINE FOR PASS2
                             001E                          <W^MAC$GL_LINENUM>
                    DC    11 0029          BRB     20$                          ;CONTINUE
           5A    00 B8    9A 002B  30$:    MOVZBL  @(R8),R10                    ;GET NEXT CHARACTER
                    68    D6 002F          INCL    (R8)                         ;POINT TO NEXT CHARACTER
                    5A    D5 0031          TSTL    R10                          ;WAS CHARACTER A NULL?
                    F6    13 0033          BEQL    30$                          ;IF EQL YES--GET NEXT CHARACTER
                             0035                                               ; (GOD ONLY KNOWS WHERE NULLS COME FROM!)
           03 6B    2F    E1 0035          BBC     #FLG$V_DLIMSTR,(R11),35$     ;ALLOW SEMICOLONS AND HYPHENS?
                  007B    31 0039          BRW     150$                         ;YES -- BYPASS OTHER TESTS.
           0E 6B    1A    E0 003C  35$:    BBS     #FLG$V_SPLALL,(R11),40$      ;BRANCH IF SHOULD NOT EVEN CONSIDER
                             0040                                               ;SEMI-COLONS
              3B    5A    91 0040          CMPB    R10,#SEMI                    ;IS CHARACTER A SEMI-COLON?
                    09    12 0043          BNEQ    40$                          ;IF NEQ NO
                 06 6B    E8 0045          BLBS    (R11),40$                    ;YES--AND ARE WE PASSING ALL CHARS.?
                             0048                                               ;(ALLCHR IS LOW BIT!!)
              5A    0D    9A 0048          MOVZBL  #CR,R10                      ;NO--CALL IT EOL
                  0069    31 004B          BRW     150$                         ;TAKE THE QUICK EXIT
              2D    5A    91 004E  40$:    CMPB    R10,#HYPHEN                  ;LINE CONTINUATION?
                    64    12 0051          BNEQ    150$                         ;IF NEQ NO
           60 6B    03    E1 0053          BBC     #FLG$V_CONT,(R11),150$       ;YES--CONTINUATIONS ALLOWED?
                    5A    DD 0057          PUSHL   R10                          ;YES--SAVE CURRENT STATE
                    68    DD 0059          PUSHL   (R8)                         ;SAVE MAC$GL_LINEPT
  7E    6B    FFFFFFFE 8F CB 005B          BICL3   #^C<FLG$M_ALLCHR>,(R11),-(SP) ;SAVE ALLCHR STATE
              6B    09    CA 0063          BICL2   #FLG$M_CONT!FLG$M_ALLCHR,(R11)   ;DON'T RECURSE ON LINES THAT HAVE
              5A    20    9A 0066          MOVZBL  #BLNK,R10                    ; ALL HYPHENS.
              20    5A    91 0069  60$:    CMPB    R10,#BLNK                    ;IS CHARACTER A BLANK?
                    05    13 006C          BEQL    70$                          ;IF EQL YES
              09    5A    91 006E          CMPB    R10,#TAB                     ;NO--IS IT A TAB?
                    05    12 0071          BNEQ    80$                          ;IF NEQ NO
                  FF8A    30 0073  70$:    BSBW    MAC$GETCHR                   ;YES--SKIP OVER SPACES AND TABS
                    F1    11 0076          BRB     60$                          ;FIND NON-BLANK,NON-TAB CHARACTER
              0D    5A    91 0078  80$:    CMPB    R10,#CR                      ;IS THIS EOL?
                    2E    12 007B          BNEQ    130$                         ;IF NEQ NO
              6B    8E    C8 007D          BISL2   (SP)+,(R11)                  ;YES--RESTORE ALLCHR FLAG
              5E    08    C0 0080          ADDL2   #2*4,SP                      ;AND SAVED CONTEXT NOT NEEDED
                  FF7A    30 0083          BSBW    MAC$GETCHR                   ;READ NEXT LINE
           04 6B    0D    E1 0086          BBC     #FLG$V_OPRND,(R11),90$       ;BRANCH IF NOT IN OPERAND FIELD
           06 6B    14    E3 008A          BBCS    #FLG$V_CHKLPND,(R11),100$ ;SET CHKL PENDING AND BRANCH IF NONE PENDI
                             008E  90$:    $INTOUT_X INT$_CHKL                  ;ENSURE ALIGNMENT OF CONTINUED LINES
              0D    5A    91 0094  100$:   CMPB    R10,#CR                      ;NULL LINE?
                    0D    13 0097          BEQL    120$                         ;IF EQL YES
              0C    5A    91 0099          CMPB    R10,#FF                      ;STILL LOOKING FOR NULL LINES
                    05    13 009C          BEQL    110$
              3B    5A    91 009E          CMPB    R10,#SEMI                    ;
                    11    12 00A1          BNEQ    140$
              5A    0D    9A 00A3  110$:   MOVZBL  #CR,R10                      ;TREAT AS EOL IF NULL
                  FF57    30 00A6  120$:   BSBW    MAC$GETCHR                   ;READ NEXT LINE
                    E9    11 00A9          BRB     100$                         ;FIND NON-NULL LINE
                             00AB  ;
                             00AB  ; NOT REALLY A CONTINUED LINE
                             00AB  ;
              6B    8E    C8 00AB  130$:   BISL2   (SP)+,(R11)                  ;RESTORE ALLCHR FLAG
                    68  8ED0 00AE          POPL    (R8)                         ;RESTORE MAC$GL_LINEPT
                    5A  8ED0 00B1          POPL    R10                          ;RESTORE CHARACTER
```

```
                   6B    08    C8  00B4    216 140$:   BISL2   #FLG$M_CONT,(R11)         ;ALLOW CONTINUATIONS AGAIN
                         58  8ED0  00B7    217 150$:   POPL    R8                        ;RESTORE R8
                               05  00BA    218         RSB                               ;RETURN WITH CHARACTER IN R10
                                   00BB    219
                                   00BB    220         .DSABL  LSB
```

```
                              00BB                .SBTTL  MAC$GETLIN GET NEXT INPUT SOURCE LINE
                              00BB
                              00BB      ;++
                              00BB      ;         THIS ROUTINE IS CALLED TO GET THE NEXT LINE OF THE CURRENT
                              00BB      ;         INPUT FILE AND PLACE IT IN MAC$AB_LINEBF.
                              00BB      ;--
                              00BB
                              00BB                .ENABL  LSB
                              00BB
                              00BB      MAC$GETLIN::
         50   0000'CF     9E  00BB                MOVAB   W^MAC$INPUT_RAB,R0        ;POINT TO THE RAB
              0000'CF     DF  00C0                PUSHAL  W^MAC$GT_SCB              ; Control block address
     00000000'GF     01   FB  00C4                CALLS   #1,G^SUM$LINE             ; Get next input line
                  6D 50   E8  00CB                BLBS    R0,40$                    ; If LBS then good read
     00000000'8F     50   D1  00CE                CMPL    R0,#RMS$_EOF              ; Was error end-of-file?
                     27   13  00D5                BEQL    8$                        ; Yes if EQL, try next file
    51   50    0C    10   EF  00D7                EXTZV   #STS$V_FAC_NO,#STS$S_FAC_NO,R0,R1 ; Get facility no
         0084 8F     51   B1  00DC                CMPW    R1,#<SUM$_NORMAL@-16>     ; SUM error?
                     0C   13  00E1                BEQL    2$                        ; Yes if EQL
 51   50   FFFFFFF8 8F    CB  00E3                BICL3   #^CSTS$M_SEVERITY,R0,R1   ; Copy and check the severity
                     CE   13  00EB                BEQL    MAC$GETLIN                ; WARNING, try again
                     0F   11  00ED                BRB     8$                        ; ERROR or FATAL, try next file
          05 6B      0E   E0  00EF      2$:       BBS     #FLG$V_P2,(R11),5$        ; Return line if Pass 2
                   00AF   30  00F3                BSBW    SUM_ERROR                 ; Generate intermediate code if Pass 1
                     C3   11  00F6                BRB     MAC$GETLIN
                              00F8      5$:
              0000'CF     D7  00F8                DECL    W^MAC$GL_LINENUM          ; Don't increment line number
                     3D   11  00FC                BRB     40$
                              00FE      8$:
                   00F5   30  00FE                BSBW    MAC$NXTINPFIL             ;ELSE TRY TO OPEN NEXT FILE
         0000'CF     01   9A  0101                MOVZBL  #1,W^MAC$GL_SRCPAG        ;RESET PAGE COUNT TO ONE
                  B2 50   E8  0106                BLBS    R0,MAC$GETLIN             ;IF THERE IS ANOTHER FILE GO READ IT
 0000'CF   444E452E 8F    D0  0109      10$:      MOVL    #^A/.END/,W^MAC$AB_LINEBF ;OOPS--NO FILE--FAKE A '.END'
         0022'CF     04   9B  0112                MOVZBW  #4,W^MAC$INPUT_RAB+RAB$W_RSZ ;WITH A SIZE OF 4 BYTES
              0000'CF     D5  0117                TSTL    W^MAC$GL_IF_LEVEL         ;IN UNFINISHED CONDITIONAL?
                     0C   15  011B                BLEQ    30$                       ;IF LEQ NO
     0004'CF     43 8F    90  011D                MOVB    #^A/C/,W^MAC$AB_LINEBF+4;YES--MAKE .END INTO .ENDC
              0022'CF     B6  0123                INCW    W^MAC$INPUT_RAB+RAB$W_RSZ ;COUNT THE CHARACTER
                     12   11  0127                BRB     40$                       ;CONTINUE
          0E 6B      0E   E0  0129      30$:      BBS     #FLG$V_P2,(R11),40$       ;ONLY MESSAGE ON PASS 1
                              012D      ;***      $INTOUT_X INT$_FNEWL              ;PRINT CONTENTS OF PREVIOUS LINE
                              012D                $INTOUT_LW INT$_WRN,-             ;TELL OF MISSING END STATEMENT
                              012D                <#MAC$_MISSINGEND,#0>
              0022'CF     3C  013B      40$:      MOVZWL  W^MAC$INPUT_RAB+RAB$W_RSZ,- ;SAVE LENGTH OF LINE
                     50       013F                                R0
         0000'CF     50   D0  0140                MOVL    R0,W^MAC$GL_LINELN        ;...
         50   0000'CF     9E  0145      50$:      MOVAB   W^MAC$AB_LINEBF,R0        ;GET ADDRESS OF LINE BUFFER
         0000'CF     50   D0  014A                MOVL    R0,W^MAC$GL_LINEPT        ;SET UP LINE POINTER
         0000'CF     50   D0  014F                MOVL    R0,W^MAC$GL_ERRPTX        ;RESET ERROR POINTER TO LINE START
              0C     60   91  0154                CMPB    (R0),#FF                  ;NEW SOURCE PAGE?
                     04   12  0157                BNEQ    60$                       ;IF NEQ NO
              0000'CF     D6  0159                INCL    W^MAC$GL_SRCPAG           ;YES--COUNT NEW PAGE
         50   0000'CF     C0  015D      60$:      ADDL2   W^MAC$GL_LINELN,R0        ;FIGURE ADDRESS OF LAST CHARACTER
              60     0D   90  0162                MOVB    #CR,(R0)                  ;STORE CR FOR END OF LINE
              0000'CF     D6  0165                INCL    W^MAC$GL_LINENUM          ;COUNT THIS LINE
          04 6B      0E   E0  0169                BBS     #FLG$V_P2,(R11),80$       ;BRANCH IF THIS IS PASS 2
              0000'CF     D6  016D                INCL    W^MAC$GL_SRC_LCNT         ;NO--COUNT LINE READ IN PASS 1
```

```
                               0171   279 80$:
                  51    D4     0171   280           CLRL    R1                            ; Initialise insert number
            13 6B   27    E0     0173   281           BBS     #FLG$V_UPDFIL,(R11),90$ ; Branch if file is being updated
     0000'CF    0000'CF    C3     0177   282           SUBL3   W^MAC$GL_LINBAS, -         ; Compute line number
                  50           017E   283                   W^MAC$GL_LINENUM,R0
            17 6B   19    E1     017F   284           BBC     #FLG$V_SEQFIL,(R11),100$ ; Branch if not sequenced file
          50    0000'CF    D0     0183   285           MOVL    W^MAC$GL_RECHDBUF,R0       ; Get sequenced line number
                  10    11     0188   286           BRB     100$
                               018A   287 90$:
          50    0018'CF    3C     018A   288           MOVZWL  W^MAC$GT_SCB+SUM_W_LINE_NO,R0 ; Get original line number
        001C'CF    02    E1     018F   289           BBC     #SUM_V_SRCUPD, -           ; Branch if line from source file
                  05           0194   290                   W^MAC$GT_SCB+SUM_B_FLAGS,100$
          51    001A'CF    3C     0195   291           MOVZWL  W^MAC$GT_SCB+SUM_W_INSERT_NO,R1 ; Get insert number
                               019A   292 100$:
        0000'CF    50    B0     019A   293           MOVW    R0,W^MAC$GW_LST_LINE       ; Save listing line number
        0000'CF    51    B0     019F   294           MOVW    R1,W^MAC$GW_LST_INST       ; and insert number
                        05     01A4   295           RSB
                               01A5   296
                               01A5   297           .DSABL  LSB
                               01A5   298 ;
                               01A5   299 ;
                               01A5   300 SUM_ERROR:
             0000'CF    D5     01A5   301           TSTL    W^MAC$GL_LINENUM           ; Error on very first line?
                  04    12     01A9   302           BNEQ    5$                         ; No if NEQ
            00 6B   29    E2     01AB   303           BBSS    #FLG$V_FIRSTLN,(R11),5$
                               01AF   304 5$:
      00848800 8F    50    D1     01AF   305           CMPL    R0,#SUM$_EDITSCLSH         ; Edits clash error?
                  22    12     01B6   306           BNEQ    4$                         ; No if NEQ
      0000001C'EF    03    E0     01B8   307           BBS     #SUM_V_SUBCLSH, -          ; Branch if not first clash
                  2F           01BF   308                   MAC$GT_SCB+SUM_B_FLAGS,10$
                  50    DD     01C0   309           PUSHL   R0
                               01C2   310           $INTOUT_X INT$_FNEWL               ; Force new line
                  50  8ED0     01C8   311           POPL    R0
             50    03    C8     01CB   312           BISL2   #STS$K_INFO,R0             ; Make and information code
                               01CE   313           $INTOUT_LW INT$_INFO,<R0,#0>       ; Information message
                  15    11     01D8   314           BRB     10$
                               01DA   315 4$:
                  50    DD     01DA   316           PUSHL   R0
                               01DC   317           $INTOUT_X INT$_FNEWL               ; Force new line
                  50  8ED0     01E2   318           POPL    R0
                               01E5   319           $INTOUT_LW INT$_WRN,<R0,#0>        ; Warning message
                               01EF   320 10$:
                               01EF   321           $INTOUT_X INT$_SUME                ; Source update merge error
                        05     01F5   322           RSB
```

```
                           01F6           .SBTTL  OPEN NEXT INPUT SOURCE FILE
                           01F6
                           01F6   ;++
                           01F6   ;       THIS ROUTINE IS CALLED TO OPEN THE NEXT INPUT FILE.
                           01F6   ;       IF THE FILE IS SUCCESSFULLY OPENED AND CONNECTED,
                           01F6   ;       RO IS SET TO 1.  IF ANOTHER INPUT FILE CANNOT BE OPENED,
                           01F6   ;       RO IS 0.
                           01F6   ;
                           01F6   ;--
                           01F6
                           01F6   MAC$NXTINPFIL:
                 3E    BB  01F6           PUSHR   #^M<R1,R2,R3,R4,R5>     ;SAVE REGISTERS USED BY MOVC5
            0000'CF    9F  01F8           PUSHAB  W^MAC$GT_SCB            ; Supply SUM control block address
      00000000'GF  01  FB  01FC           CALLS   #1,G^SUM$CLOSE          ; Close any update files
                           0203           $DISCONNECT RAB=W^MAC$INPUT_RAB ;DISCONNECT THE RECORD ACCESS
       50   0000'CF    D0  020E           MOVL    W^MAC$GL_CURINFDB,R0    ;POINT TO CURRENT FDB
                           0213           $CLOSE  FAB=8(R0)               ;CLOSE THE INPUT FILE
            0000'CF    D0  021D           MOVL    W^MAC$GL_LINENUM,-      ;SET LINE BASE TO HIGHEST LINE NUMBER
            0000'CF        0221                           @^MAC$GL_LINBAS ; ...
  27   20   6E   00    2C  0224           MOVC5   #0,(SP),#^A/ /,#39,-    ;BLANK FILL THE RESULTANT FILENAME
            0000'CF        0229                           W^MAC$AB_SBT_FILE
       50   0000'CF    D0  022C           MOVL    W^MAC$GL_CURINFDB,R0    ;GET ADDRESS OF FDB WE JUST CLOSED
       0004'CF   50    D1  0231           CMPL    R0,W^MAC$GL_INPQUE+4    ;WAS IT THE LAST FDB
                 09    13  0236           BEQL    20$                     ;IF EQL YES
            50   60    D0  0238           MOVL    (R0),R0                 ;NO--LINK TO NEXT FDB
              FDC2'    30  023B           BSBW    MAC$OPEN_INPUT          ;OPEN NEXT INPUT FILE
                 3E    BA  023E           POPR    #^M<R1,R2,R3,R4,R5>     ;RESTORE REGISTERS
                 05        0240           RSB                             ;RETURN WITH LBS IN R0
                           0241   ;
                           0241   ; NO MORE INPUT FILES
                           0241   ;
                 50    D4  0241   20$:    CLRL    R0                      ;INDICATE NO MORE FILES
                 3E    BA  0243           POPR    #^M<R1,R2,R3,R4,R5>
                 05        0245           RSB
```

```
                             0246   359          .SBTTL  STAT4           SWITCH INPUT TO MACRO TEXT
                             0246   360
                             0246   361 ;++
                             0246   362 ; FUNCTIONAL DESCRIPTION:
                             0246   363 ;
                             0246   364 ;        THIS SEMANTIC ROUTINE SWITCHES THE POINTER TO THE CURRENT
                             0246   365 ;        INPU BLOCK TO POINT TO THE MACRO BEING EXPANDED.  FIRST THE
                             0246   366 ;        REAL MACRO ARGUMENTS ARE SCANNED AND AN INPUT BLOCK IS CREATED.
                             0246   367 ;        THEN MAC$GETCHR WILL READ CHARACTERS FROM THE MACRO TEXT
                             0246   368 ;        EXPANDER MAC$GET_MAC_LIN.
                             0246   369 ;
                             0246   370 ;--
                             0246   371
                         00000000   372          .PSECT  MAC$RO_CODE_P1,NOWRT,GBL,LONG
                             0000   373 STAT4::                                  ;STATEMENT = MACTXT
        0E 0005'CF      E8   0000   374          BLBS    W^LST$G_MACROCALL+SYM$L_VAL,10$ ;BRANCH IF LISTING MACRO CALLS
                             0005   375          $INTOUT_LW INT$_SETLONG,<#0,#MAC$GL_LIST_IT> ;NO--SEND FLAG TO PASS 2
    56   0000'CF47      D0   0013   376 10$:     MOVL    W^MAC$AL_VALSTACK[R7],R6 ;GET PTR TO MACRO MNB
       55   00'8F       9A   0019   377          MOVZBL  #CRF$K_REF,R5           ;THIS IS A REFERENCE
             FFE0'      30   001D   378          BSBW    MAC$CREF_MACRO          ;CROSS REFERENCE IF CREFFING
             05 A6      D5   0020   379          TSTL    MNB$L_TXTP(R6)          ;IS THERE ANY TEXT
                1D      13   0023   380          BEQL    20$                     ;IF EQL NO--TAKE THE QUICK OUT
      0000'CF   56      D0   0025   381          MOVL    R6,W^MAC$GL_MACPTR      ;SAVE POINTER TO MACRO MNB
   0000'CF    18 A6     D0   002A   382          MOVL    MNB$L_ARGP(R6),W^MAC$GL_KEYMAC ;SET PTR TO KEYWORD ARGS (IF ANY)
             FFCD'      30   0030   383          BSBW    MAC$GET_ARGS            ;SCAN REAL MACRO ARGS
                             0033   384                                          ;AND SET UP INPUT BLOCK
           0000'CF      D4   0033   385          CLRL    W^MAC$GL_KEYMAC         ;CLEAR POINTER TO KEYWORD ARGS
 0000'CF   0000'CF      D0   0037   386          MOVL    W^MAC$GL_BLKPTR,W^MAC$GL_INPUTP ;POINT TO NEW INPUT BLOCK
        00 6B   10      E3   003E   387          BBCS    #FLG$V_MACTXT,(R11),20$ ;FLAG READING MACRO TEXT
           5A   0D      9A   0042   388 20$:     MOVZBL  #CR,R10                 ;FORCE READING OF NEW LINE
                        05   0045   389 30$:     RSB
```

```
                                    0046              .SBTTL  MEXIT           MACRO EXIT ROUTINE
                                    0046
                                    0046     ;++
                                    0046     ; FUNCTIONAL DESCRIPTION:
                                    0046     ;
                                    0046     ;        THIS ROUTINE POPS THE INPUT CONTEXT ONE LEVEL TO EFFECT AN
                                    0046     ;        EXIT FROM A MACRO OR REPEAT-TYPE MACRO
                                    0046     ;
                                    0046     ;--
                                    0046
                                    0046     MEXIT::                                           ;DIRECTIVE = KMEXIT
                                    0046
                                    0046              .SBTTL  MAC$POP_INPUT   POP INPUT CONTEXT UP A LEVEL
                                    0046
                                    0046     ;++
                                    0046     ; FUNCTIONAL DESCRIPTION:
                                    0046     ;
                                    0046     ;        THIS ROUTINE POPS THE INPUT CONTEXT A LEVEL.  ALL PAGES
                                    0046     ;        ALLOCATED TO THE CURRENT INPUT BLOCK ARE DEALLOCATED.
                                    0046     ;
                                    0046     ;--
                                    0046
                                    0046     MAC$POP_INPUT::
                          56    DD  0046              PUSHL   R6                          ;SAVE R6
                 56   0000'CF   D0  0048              MOVL    W^MAC$GL_INPUTP,R6          ;GET POINTER TO CURRENT INPUT BLOCK
                    1B 6B   10  E1  004D              BBC     #FLG$V_MACTXT,(R11),10$     ;BRANCH IF NOT READING MACRO TEXT
                  50    14  A6  D0  0051              MOVL    INP$L_RPTCNT(R6),R0         ;YES--GET REPEAT COUNT (OR MACRO FLAG)
                            15  18  0055              BGEQ    10$                         ;IF GEQ NO NEED TO GO ANY FARTHER
                            50  D6  0057              INCL    R0                          ;SEE IF IT WAS REPEAT THAT WENT TO 0
                            11  13  0059              BEQL    10$                         ;IF EQL YES
                                    005B                                                  ;NO--IT WAS MACRO REDEFINING ITSELF
  56   14 A6   80000000 8F      CB  005B              BICL3   #^X80000000,INP$L_RPTCNT(R6),R6 ;YES--GET ADDRESS OF MNB
                         FF99'  30  0064              BSBW    MAC$DEL_MAC_DEF             ;AND DELETE THE MNB AND ASSOCIATES
                 56   0000'CF   D0  0067              MOVL    W^MAC$G[_INPUTP,R6          ;RE-GET THE INPUT BLOCK ADDRESS
                       50   66  D0  006C     10$:     MOVL    INP$L_LINK(R6),R0           ;POP INPUT CONTEXT
                  0000'CF   50  D0  006F              MOVL    R0,W^MAC$GL_INPUTP          ;...
              00000000'8F   50  D1  0074              CMPL    R0,#MAC$GL_PRMINBL          ;READING FROM SOURCE FILE?
                            04  12  007B              BNEQ    20$                         ;IF NEQ NO
                    00 6B   10  E5  007D              BBCC    #FLG$V_MACTXT,(R11),20$ ;YES--CLEAR MACTXT FLAG
               0000'CF  10  A6  D0  0081     20$:     MOVL    INP$L_IFVAL(R6),W^MAC$GL_IF_VALUE ;POP IF VALUE
               0000'CF  0C  A6  D0  0087              MOVL    INP$L_IFLVL(R6),W^MAC$GL_IF_LEVEL ;POP IF LEVEL
                    50      56  D0  008D              MOVL    R6,R0                       ;COPY INPUT BLOCK ADDRESS
                  56    18  A6  D0  0090              MOVL    INP$L_PAGP(R6),R6           ;LINK TO ANY OTHER PAGES
                         FF69'  30  0094              BSBW    MAC$DEA_1_PAGE              ; Deallocate block (always 1 page)
                            06  11  0097              BRB     40$                         ;GO DEALLOCATE INPUT BLOCK AND REST
                                    0099                                                  ;OF INPUT CONTEXT PAGES
                                    0099     30$:
                    56      66  D0  0099              MOVL    MXB$L_LINK(R6),R6           ; Link to possible next page
                         FF61'  30  009C              BSBW    MAC$DEAL_BLOCK              ; Deallocate memory block
                                    009F     40$:
                    50      56  D0  009F              MOVL    R6,R0                       ;POINT R0 FOR NEXT DEALLOCATION
                            F5  12  00A2              BNEQ    30$                         ;IF NEQ GO DO IT
                    5A      0D  9A  00A4              MOVZBL  #CR,R10                     ;FORCE NEW LINE
                            56 8ED0 00A7              POPL    R6                          ;RESTORE R6
                                05  00AA              RSB
                         00AB          .SBTTL  STATEMENTS      END-OF-LINE CLEANUP
                         00AB
                         00AB  ;++
                         00AB  ;       AFTER EACH STATEMENT SOME FLAGS NEED TO BE RESET.
                         00AB  ;--
                         00AB
                         00AB  ;MTXT1::                                        ;MARS_TEXT = STATEMENT
                         00AB  ;MTXT2::                                        ;MARS_TEXT = MARS_TEXT STATEMENT
                         00AB  MTXT3::                                 ;MARS_TEXT = DEOL
                         00AB  MTXT4::                                 ;MARS_TEXT = MARS_TEXT DEOL
                         00AB  MTEXT:
                         00AB                                          ;BEFORE FOR THIS LINE
                         00AB          $INTOUT_X INT$_CHKL             ;ALIGN LISTING AND SOURCE
     00000046 8F    C8   00B1          BISL2   #FLG$M_BOL!FLG$M_EVALEXPR!FLG$M_COMPEXPR,- ;RESET BEGINNING OF LINE
              6B         00B7                          (R11)           ;AND EVALUATE FLAGS
         0000'CF    D4   00B8          CLRL    W^MAC$GL_ABSFLAG        ;RESET ABSOLUTE FLAG
         0000'CF    D4   00BC          CLRL    W^MAC$GL_PRMSEG         ;NO EXPRESSION PSECT YET
         0000'CF    D4   00C0          CLRL    W^MAC$GL_MOPPTR         ;CLEAR POINTER TO OPERAND MODE BYTES
6B   01802000 8F    CA   00C4          BICL2   #FLG$M_OPRND!FLG$M_IFSTAT!FLG$M_NOREF,(R11) ;NOT  IN OPERAND FIELD
                         00CB                                                  ;AND NOT IN AN IF
                         00CB                                                  ;AND ALLOW PRMSYM TO REF SYMBOLS
     0000'CF  02    90   00CB          MOVB    #RDX$V_DECIMAL,W^MAC$GB_RDXNDX ;RESET TO DECIMAL RADIX
                    05   00D0          RSB
                         00D1
                         00D1          .END
```

```
$$.TMP1                          = 00000001              FLG$M_NEWPND                     = 00000400
$$.TMP2                          = 000000A0              FLG$M_NOREF                      = 01000000
$COUNT                           = 0000003B              FLG$M_NTYPEPC                    = 00000020
ARG$K_SIZE                       = 000003E8              FLG$M_NULCHR                     = 00040000
AUD$K_SIZE                       = 00000010              FLG$M_OBJXST                     = 00200000
BIT...                           = 00000005              FLG$M_OPNDCHK                    = 00000100
BLNK                             = 00000020              FLG$M_OPRND                      = 00002000
CHR$M_COMMA_CR                   = 00000020              FLG$M_OPTVFLIDX                  = 00001000
CHR$M_ILL_CHR                    = 00000040              FLG$M_ORDLST                     = 00020000
CHR$M_NUM_BER                    = 00000010              FLG$M_P2                         = 00004000
CHR$M_SPA_MSK                    = 00000001              FLG$M_RPTIRP                     = 10000000
CHR$M_SYM_CH1                    = 00000008              FLG$M_SEQFIL                     = 02000000
CHR$M_SYM_CHR                    = 00000004              FLG$M_SKAN                       = 00008000
CHR$M_SYM_DLM                    = 00000002              FLG$M_SPECOP                     = 00000004
CHR$V_COMMA_CR                   = 00000005              FLG$M_SPLALL                     = 04000000
CHR$V_CVTLWC                     = 00000061              FLG$M_STOIMF                     = 00040000
CHR$V_ILL_CHR                    = 00000006              FLG$M_SYM2COL                    = 00000400
CHR$V_NOCVT                      = 0000007F              FLG$M_TOCFLG                     = 00080000
CHR$V_NUM_BER                    = 00000004              FLG$M_UPAFLG                     = 00000010
CHR$V_SPA_MSK                    = 00000000              FLG$M_UPDFIL                     = 00000080
CHR$V_SYM_CH1                    = 00000003              FLG$M_UPMARG                     = 00000040
CHR$V_SYM_CHR                    = 00000002              FLG$M_XCRF                       = 80000000
CHR$V_SYM_DLM                    = 00000001              FLG$V_ALLCHR                     = 00000000
CNT                              = 00000002              FLG$V_BOL                        = 00000001
CR                               = 0000000D              FLG$V_CHKLPND                    = 00000014
CRF$K_REF                          ********   X    04    FLG$V_COMPEXPR                   = 00000002
ERR                              = 00000000              FLG$V_CONT                       = 00000003
FF                               = 0000000C              FLG$V_CRF                        = 0000001E
FLG$M_ALLCHR                     = 00000001              FLG$V_CRSEEN                     = 00000020
FLG$M_BOL                        = 00000002              FLG$V_DATRPT                     = 00000004
FLG$M_CHKLPND                    = 00100000              FLG$V_DBGOUT                     = 0000002E
FLG$M_COMPEXPR                   = 00000004              FLG$V_DLIMSTR                    = 0000002F
FLG$M_CONT                       = 00000008              FLG$V_ENDMCH                     = 00000005
FLG$M_CRF                        = 40000000              FLG$V_EVALEXPR                   = 00000006
FLG$M_CRSEEN                     = 00000001              FLG$V_EXPOPT                     = 00000007
FLG$M_DATRPT                     = 00000010              FLG$V_EXTERR                     = 00000030
FLG$M_DBGOUT                     = 00004000              FLG$V_EXTWRN                     = 00000031
FLG$M_DLIMSTR                    = 00008000              FLG$V_FIRSTLN                    = 00000029
FLG$M_ENDMCH                     = 00000020              FLG$V_IFSTAT                     = 00000017
FLG$M_EVALEXPR                   = 00000040              FLG$V_IIF                        = 00000016
FLG$M_EXPOPT                     = 00000080              FLG$V_INSERT                     = 00000008
FLG$M_EXTERR                     = 00010000              FLG$V_IRPC                       = 0000001D
FLG$M_EXTWRN                     = 00020000              FLG$V_LEXOP                      = 00000021
FLG$M_FIRSTLN                    = 00000200              FLG$V_LSTXST                     = 00000009
FLG$M_IFSTAT                     = 00800000              FLG$V_MAC2COL                    = 0000002B
FLG$M_IIF                        = 00400000              FLG$V_MACL                       = 0000000B
FLG$M_INSERT                     = 00000100              FLG$V_MACLTB                     = 0000001B
FLG$M_IRPC                       = 20000000              FLG$V_MACTXT                     = 00000010
FLG$M_LEXOP                      = 00000002              FLG$V_MEBLST                     = 0000000C
FLG$M_LSTXST                     = 00000200              FLG$V_MOREARG                    = 0000002D
FLG$M_MAC2COL                    = 00000800              FLG$V_MOREINP                    = 00000023
FLG$M_MACL                       = 00000800              FLG$V_NEWPND                     = 0000000A
FLG$M_MACLTB                     = 08000000              FLG$V_NOREF                      = 00000018
FLG$M_MACTXT                     = 00010000              FLG$V_NTYPEPC                    = 00000025
FLG$M_MEBLST                     = 00001000              FLG$V_NULCHR                     = 00000032
FLG$M_MOREARG                    = 00002000              FLG$V_OBJXST                     = 00000015
FLG$M_MOREINP                    = 00000008              FLG$V_OPNDCHK                    = 00000028
```

```
FLG$V_OPRND                    = 0000000D              INT$_PRT                       = 0000001B
FLG$V_OPTVFLIDX                = 0000002C              INT$_PSECT                     = 0000001C
FLG$V_ORDLST                   = 00000011              INT$_REDEF                     = 0000001D
FLG$V_P2                       = 0000000E              INT$_REF                       = 0000001E
FLG$V_RPTIRP                   = 0000001C              INT$_REST                      = 0000001F
FLG$V_SEQFIL                   = 00000019              INT$_SAME                      = 00000009
FLG$V_SKAN                     = 0000000F              INT$_SAVE                      = 00000020
FLG$V_SPECOP                   = 00000022              INT$_SBTTL                     = 00000021
FLG$V_SPLALL                   = 0000001A              INT$_SETFLAG                   = 00000022
FLG$V_STOIMF                   = 00000012              INT$_SETLONG                   = 00000023
FLG$V_SYM2COL                  = 0000002A              INT$_SPIC                      = 00000024
FLG$V_TOCFLG                   = 00000013              INT$_SPID                      = 00000025
FLG$V_UPAFLG                   = 00000024              INT$_STIB                      = 00000026
FLG$V_UPDFIL                   = 00000027              INT$_STIL                      = 00000028
FLG$V_UPMARG                   = 00000026              INT$_STIW                      = 00000027
FLG$V_XCRF                     = 0000001F              INT$_STKEPT                    = 00000029
HASHSZ                         = 0000007F              INT$_STKG                      = 0000002A
HYPHEN                         = 0000002D              INT$_STKL                      = 0000002B
INP$B_ARGCT                      0000001C              INT$_STKPC                     = 0000002C
INP$K_BLKSIZ                     00000021              INT$_STKS                      = 0000002D
INP$K_BUFSIZ                   = 000003E8              INT$_STOB                      = 00000034
INP$K_IRPSIZ                     0000003C              INT$_STOL                      = 0000002E
INP$L_ARGS                       0000001D              INT$_STOW                      = 00000035
INP$L_GETL                       00000008              INT$_STRB                      = 0000002F
INP$L_IFLVL                      0000000C              INT$_STRL                      = 00000031
INP$L_IFVAL                      00000010              INT$_STRSB                     = 00000032
INP$L_LINK                       00000000              INT$_STRSW                     = 00000033
INP$L_NXTL                       00000004              INT$_STRW                      = 00000030
INP$L_PAGP                       00000018              INT$_STSB                      = 00000036
INP$L_RPTCNT                     00000014              INT$_STSW                      = 00000037
INT$K_BUFSIZ                   = 000013F4              INT$_SUB                       = 0000000A
INT$K_BUFWRN                   = 00001390              INT$_SUME                      = 00000039
INT$_ADD                       = 00000001              INT$_WRN                       = 00000038
INT$_AND                       = 00000002              INT$_XOR                       = 0000000B
INT$_ASH                       = 00000003              LST$G_MACROCALL                  ********   X   04
INT$_ASN                       = 0000000C              LST$K_BUFSIZ                   = 00000086
INT$_AUGPC                     = 0000000D              LST$K_L_P_PAGE                 = 0000003C
INT$_BDST                      = 0000000E              LST$K_TITLE_SIZ                = 00000028
INT$_CHKL                      = 0000000F              MAB$B_ARGNO                      00000005
INT$_DIV                       = 00000004              MAB$B_NAME                       00000004
INT$_END                       = 00000010              MAB$K_BLKSIZ                     0000000C
INT$_EPT                       = 00000011              MAB$L_DVPTR                      00000008
INT$_ERR                       = 00000012              MAB$L_LINK                       00000000
INT$_ETX                       = 00000013              MAB$W_DVLEN                      00000006
INT$_FNEWL                     = 00000014              MAC$AB_LINEBF                    ********   X   03
INT$_ILG                       = 00000000              MAC$AB_SBT_FILE                  ********   X   03
INT$_INFO                      = 0000003A              MAC$AL_VALSTACK                  ********   X   04
INT$_LGLAB                     = 00000015              MAC$CREF_MACRO                   ********   X   04
INT$_MACL                      = 00000016              MAC$DEAL_BLOCK                   ********   X   04
INT$_MUL                       = 00000005              MAC$DEA_T_PAGE                   ********   X   04
INT$_NEG                       = 00000006              MAC$DEL_MAC_DEF                  ********   X   04
INT$_NEWL                      = 00000017              MAC$GB_RDXNDX                    ********   X   04
INT$_NEWP                      = 00000018              MAC$GETCHR                       00000000 RG    03
INT$_NOT                       = 00000007              MAC$GETLIN                       000000BB RG    03
INT$_OP                        = 00000019              MAC$GET_ARGS                     ********   X   04
INT$_OR                        = 00000008              MAC$GL_ABSFLAG                   ********   X   04
INT$_PRIL                      = 0000001A              MAC$GL_BLKPTR                    ********   X   04
```

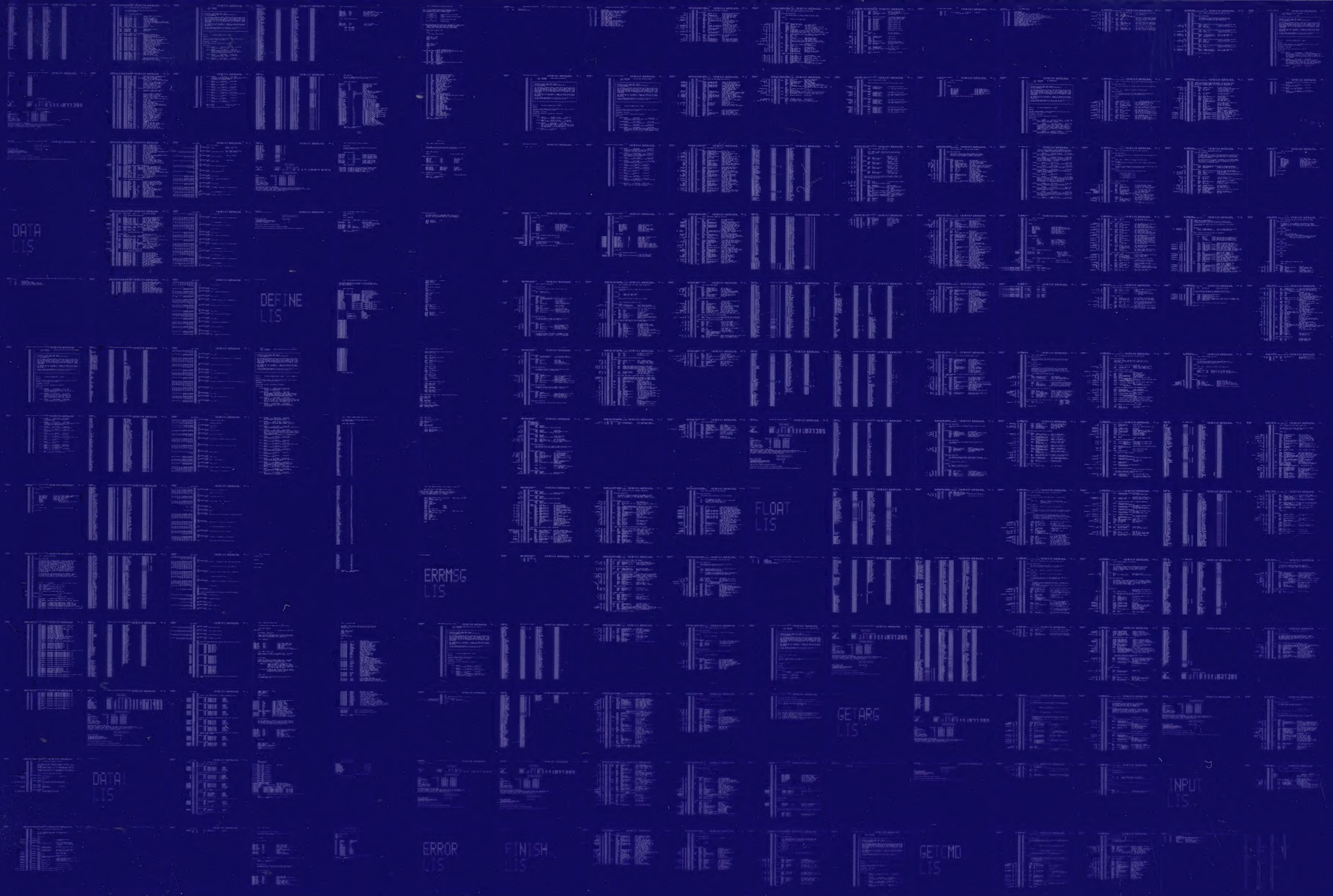
```
MAC$GL_CURINFDB                   ********   X    03         PSC$K_BLKSIZ                     00000013
MAC$GL_ERRPTX                     ********   X    03         PSC$K_NO_OPTNS                 = 0000000A
MAC$GL_IF_LEVEL                   ********   X    03         PSC$L_CURLOC                     0000000F
MAC$GL_IF_VALUE                   ********   X    04         PSC$L_LINK                       00000000
MAC$GL_INPQUE                     ********   X    03         PSC$L_MAXLGTH                    00000005
MAC$GL_INPUTP                     ********   X    03         PSC$M_ABS                      = FFFFFFF7
MAC$GL_KEYMAC                     ********   X    04         PSC$M_ALIGNFLG                 = 00004000
MAC$GL_LINBAS                     ********   X    03         PSC$M_ALLOPTNS                 = 000003FF
MAC$GL_LINELN                     ********   X    03         PSC$M_BYTE                     = 00004000
MAC$GL_LINENUM                    ********   X    03         PSC$M_CON                      = FFFFFFFB
MAC$GL_LINEPT                     ********   X    03         PSC$M_DEFAULT                  = 000001C8
MAC$GL_LIST_IT                    ********   X    04         PSC$M_EXE                      = 000000C0
MAC$GL_MACPTR                     ********   X    04         PSC$M_GBL                      = 00000010
MAC$GL_MOPPTR                     ********   X    04         PSC$M_LCL                      = FFFFFFEF
MAC$GL_PRMINBL                    ********   X    04         PSC$M_LIB                      = 00000002
MAC$GL_PRMSEG                     ********   X    04         PSC$M_LONG                     = 00004800
MAC$GL_RECHDBUF                   ********   X    03         PSC$M_NOEXE                    = FFFFFFBF
MAC$GL_SRCPAG                     ********   X    03         PSC$M_NOPIC                    = FFFFFFFE
MAC$GL_SRC_LCNT                   ********   X    03         PSC$M_NORD                     = FFFFFF7F
MAC$GT_SCB                        ********   X    03         PSC$M_NOSHR                    = FFFFFFDF
MAC$GW_LST_INST                   ********   X    03         PSC$M_NOVEC                    = FFFFFDFF
MAC$GW_LST_LINE                   ********   X    03         PSC$M_NOWRT                    = FFFFFEFF
MAC$INPUT_RAB                     ********   X    03         PSC$M_OVR                      = 00000004
MAC$INTERR_2_LW                   ********   X    03         PSC$M_PAGE                     = 00006400
MAC$INTOUT_2_LW                   ********   X    03         PSC$M_PIC                      = 00000001
MAC$INTOUT_WD                     ********   X    03         PSC$M_QUAD                     = 00004C00
MAC$INTOUT_X                      ********   X    03         PSC$M_RD                       = 00000080
MAC$NXTINPFIL                     000001F6 R      03         PSC$M_REL                      = 00000008
MAC$OPEN_INPUT                    ********   X    03         PSC$M_SHR                      = 00000020
MAC$POP_INPUT                     00000046 RG     04         PSC$M_USR                      = FFFFFFFD
MAC$_MISSINGEND                 = 007D8828                   PSC$M_VEC                      = 00000200
MAC_SUBSYS                      = 0000007D                   PSC$M_WORD                     = 00004400
MEXIT                             00000046 RG     04         PSC$M_WRT                      = 00000180
MNB$B_ARGCT                       00000017                   PSC$S_ALIGNMENT                = 00000004
MNB$B_NAME                        00000004                   PSC$V_ALIGNFLG                 = 0000000E
MNB$K_BLKSIZ                      0000001C                   PSC$V_ALIGNMENT                = 0000000A
MNB$L_ARGP                        00000018                   PSC$V_EXE                      = 00000006
MNB$L_CRSYMF                      00000013                   PSC$V_GBL                      = 00000004
MNB$L_LINK                        00000000                   PSC$V_LIB                      = 00000001
MNB$L_PAGC                        0000000F                   PSC$V_OVR                      = 00000002
MNB$L_PAGP                        0000000B                   PSC$V_PIC                      = 00000000
MNB$L_TXTP                        00000005                   PSC$V_RD                       = 00000007
MNB$W_FLAG                        00000009                   PSC$V_REL                      = 00000003
MTEXT                             000000AB R      04         PSC$V_SHR                      = 00000005
MTXT3                             000000AB RG     04         PSC$V_VEC                      = 00000009
MTXT4                             000000AB RG     04         PSC$V_WRT                      = 00000008
MXB$K_BLKSIZ                      00000008                   PSC$W_FLAG                       00000009
MXB$L_LINK                        00000000                   PSC$W_OPTIONS                    0000000D
MXB$L_PAGES                       00000004                   RAB$W_RSZ                      = 00000022
OBJ$K_BUFSIZ                    = 00000200                   RDX$V_BINARY                   = 00000000
OPF$M_LASTOPR                   = 00002000                   RDX$V_DECIMAL                  = 00000002
OPF$M_OPTEXP                    = 00001000                   RDX$V_DOUBLE                   = 00000005
OPF$V_LASTOPR                   = 0000000D                   RDX$V_FLOAT                    = 00000004
OPF$V_OPTEXP                    = 0000000C                   RDX$V_GFLOAT                   = 00000006
PSC$B_NAME                        00000004                   RDX$V_HEX                      = 00000003
PSC$B_SEG                         0000000C                   RDX$V_HFLOAT                   = 00000007
PSC$B_UNUSED                      0000000B                   RDX$V_OCTAL                    = 00000001
```

```
REG$_PC                         = 0000000F                SYM$V_ABS                       = 00000004
RMS$_EOF                          ********   X     03     SYM$V_ASN                       = 00000008
SEMI                            = 0000003B                SYM$V_CRFO                      = 0000000D
SIZ...                          = 00000001                SYM$V_DEBUG                     = 00000005
STAT4                             00000000 RG      04     SYM$V_DEF                       = 00000000
STB$K_PG_MISS                   = 0000000A                SYM$V_DELMAC                    = 00000009
STS$K_INFO                      = 00000003                SYM$V_EPT                       = 00000009
STS$M_SEVERITY                  = 00000007                SYM$V_EXTRN                     = 00000003
STS$S_FAC_NO                    = 0000000C                SYM$V_GLOBL                     = 00000002
STS$V_FAC_NO                    = 00000010                SYM$V_LOCAL                     = 00000006
SUM$CLOSE                         ********   X     03     SYM$V_ODBG                      = 0000000A
SUM$LINE                          ********   X     03     SYM$V_REF                       = 00000007
SUM$_EDITSCLSH                  = 00848800                SYM$V_RELPSECT                  = 0000000B
SUM$_NORMAL                     = 00848001                SYM$V_SUPR                      = 0000000E
SUM_B_FLAGS                       0000001C                SYM$V_WEAK                      = 00000001
SUM_ERROR                         000001A5 R       03     SYM$V_XCRF                      = 0000000C
SUM_K_BLN                         0000001D                SYM$W_FLAG                        00000009
SUM_L_ISDATA                      00000004                SYS$CLOSE                         ********  GX    03
SUM_L_STS                         00000000                SYS$DISCONNECT                    ********  GX    03
SUM_M_AUDIT                     = 00000001                TAB                             = 00000009
SUM_M_AUDITNEW                  = 00000002                X1                              = 00000400
SUM_M_DELETE                    = 00000010                X2                              = 0000000F
SUM_M_SRCUPD                    = 00000004
SUM_M_SUBCLSH                   = 00000008
SUM_Q_AUDDS                       00000008
SUM_Q_FILESP                      00000010
SUM_V_AUDIT                     = 00000000
SUM_V_AUDITNEW                  = 00000001
SUM_V_DELETE                    = 00000004
SUM_V_SRCUPD                    = 00000002
SUM_V_SUBCLSH                   = 00000003
SUM_W_INSERT_NO                   0000001A
SUM_W_LINE_NO                     00000018
SYM$B_NAME                        00000004
SYM$B_SEG                         0000000C
SYM$B_TOKEN                       0000000B
SYM$K_BLKSIZ                      0000000D
SYM$K_MAXLEN                    = 0000001F
SYM$K_TWOCOL                    = 00000010
SYM$L_LINK                        00000000
SYM$L_VAL                         00000005
SYM$M_ABS                       = 00000010
SYM$M_ASN                       = 00000100
SYM$M_CRFO                      = 00002000
SYM$M_DEBUG                     = 00000020
SYM$M_DEF                       = 00000001
SYM$M_DELMAC                    = 00000200
SYM$M_EPT                       = 00000200
SYM$M_EXTRN                     = 00000008
SYM$M_GLOBL                     = 00000004
SYM$M_LOCAL                     = 00000040
SYM$M_ODBG                      = 00000400
SYM$M_REF                       = 00000080
SYM$M_RELPSECT                  = 00000800
SYM$M_SUPR                      = 00004000
SYM$M_WEAK                      = 00000002
SYM$M_XCRF                      = 00001000
```

```
                                          +----------------+
                                          ! Psect synopsis !
                                          +----------------+

PSECT name                     Allocation          PSECT No.  Attributes
----------                     ----------          ---------  ----------
.  ABS  .                      00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
. BLANK .                      00000000  (    0.)  01 (  1.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$ABS$                          0000003C  (   60.)  02 (  2.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
MAC$RO_CODE_P15                00000246  (  582.)  03 (  3.)  NOPIC   USR   CON   REL   GBL NOSHR   EXE   RD   NOWRT NOVEC LONG
MAC$RO_CODE_P1                 000000D1  (  209.)  04 (  4.)  NOPIC   USR   CON   REL   GBL NOSHR   EXE   RD   NOWRT NOVEC LONG

                                        +------------------------+
                                        ! Performance indicators !
                                        +------------------------+

Phase                      Page faults    CPU Time       Elapsed Time
-----                      -----------    --------       ------------
Initialization                     29    00:00:00.02    00:00:01.67
Command processing                107    00:00:00.48    00:00:04.12
Pass 1                            263    00:00:04.89    00:00:24.79
Symbol table sort                   0    00:00:00.65    00:00:02.10
Pass 2                            102    00:00:01.14    00:00:04.59
Symbol table output                50    00:00:00.23    00:00:00.43
Psect synopsis output               2    00:00:00.02    00:00:00.02
Cross-reference output              0    00:00:00.00    00:00:00.00
Assembler run totals              555    00:00:07.43    00:00:37.72

The working set limit was 1500 pages.
42445 bytes (83 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 652 non-local and 38 local symbols.
471 source lines were read in Pass 1, producing 22 object records in Pass 2.
25 pages of virtual memory were used to define 23 macros.

                                        +--------------------------+
                                        ! Macro library statistics !
                                        +--------------------------+

Macro library name                        Macros defined
------------------                        --------------
_$255$DUA28:[SHRLIB]SUM.MLB;1                   3
_$255$DUA28:[MACRO.OBJ]MACRO.MLB;1             12
_$255$DUA28:[SYSLIB]STARLET.MLB;2               9
TOTALS (all libraries)                         24

827 GETS were required to define 24 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:INPUT/OBJ=OBJ$:INPUT MSRC$:INPUT/UPDATE=(ENH$:INPUT)+LIB$:MACRO/LIB+SHRLIB$:SUM/LIB
```

0225 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0226 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

MACLIB
LIS

MAIN
LIS

LINK
LIS

INTOUT
LIS

IODAT
LIS

MACMSG
LIS

P2ACT1
LIS

MACLIN
LIS

LRPTAB
LIS

OHDOUT
LIS

MACDEF
LIS

MACSUB
LIS